

PHPExcelによるExcelファイルの読み込み

岩田 員典（愛知大学経営学部）

1. はじめに

本学の新入生向けの授業である情報リテラシー入門・応用では、学生がWordやExcelの多数の問題に取り組めるように自動採点システムを独自に構築・運用している^{1), 2)}。論文¹⁾においてはExcelの採点にはPerlのライブラリであるSpreadsheet::ParseExcel³⁾を利用していた。このライブラリはOffice 2003まで用いられていたバイナリ形式のExcelファイルを読み込むための物である。Office 2007以降はファイル形式がOffice Open XML⁴⁾に変更されたため、論文²⁾ではPHPのライブラリであるPHPExcel⁵⁾を用いることになった^{a)}。そこで、本稿ではPHPExcelを利用したExcelファイルの読み込みについて解説する。

2. PHPとは

正式な名称はPHP: Hypertext Preprocessor⁶⁾であり、通常PHPと省略される。PHPの部分は“Personal Home Page”に由来していると言われている。

^{a)} Office Open XMLに対応しているPerlのライブラリとしてはSpreadsheet-XLSXがある

PHPは動的にHTMLデータを生成することによって、動的なWebページを実現することを主な目的としている。PHPといった場合にはプログラミング言語かその言語プロセッサを指す。PHPはスクリプト言語であり、サーバサイドで動作するアプリケーションの開発に用いられる。通常は、Webサーバ上で動作し、サーバ上でPHPスクリプトの文書が要求されるたびに、そのスクリプトが実行される。そして、その結果をウェブブラウザに対して送信することで動的なWebページを実現している。C言語に似た平易な文法のため幅広く用いられている。また、本稿で取り上げるPHPExcelのような多くのフレームワークやライブラリが存在し、様々なWebアプリケーションの開発がPHPによって容易になる。

2.1 PHPのインストール

PHPExcelを利用するには、PHPをインストールする必要がある。

2.1.1 Windowsへのインストール

Window版のバイナリを参考文献⁷⁾で示したURLからダウンロードすること

が出来る。以下のように大きく分けて4種類存在するzipファイルから適切な物をダウンロードする。

- 32bit (x86) Non Thread Safe
- 32bit (x86) Thread Safe
- 64bit (x64) Non Thread Safe
- 64bit (x64) Thread Safe

ビット (bit) 数は環境にあわせて選択する。そして、Non Thread SafeかThread SafeかはWebサーバの種類とVersionによる。Apache2以降ならばThread Safeを利用し、IISやApache1ならばNon Thread Safeを利用する。ただし、本稿で説明するPHPExcelを利用するだけならビット数があっていればよい。

ダウンロードしたzipファイルを好みの場所に展開する。展開したフォルダに存在するphp.exeをコマンドプロンプト等で実行できれば、そこにパスを通すことでインストールは完了する。尚、環境によっては実行に「Visual Studio 2012 更新プログラム4のVisual C++再頒布可能パッケージ」などのライブラリが必要となることがある。

2.1.2 その他のOSへのインストール

Mac OS X 10.0.0以降では標準でインストールされているので、新しいバージョンを利用するのでなければターミナルなどからすぐに利用することが出来

る。2015年10月の時点ではMac OS X 10.11.1 (El Capitan) にはPHP 5.5.29がインストールされている。

その他のUnixシステムでは、OS毎のパッケージマネージャーを利用するかソースファイルをダウンロードしてインストールすることになる。詳細は参考文献⁶⁾のDocumentationに書かれている。

また、サーバサイドで利用する場合にはApache等のWebサーバの設定も必要であり、その際には同じく参考文献⁶⁾に設定方法が記載されている。

3. PHPExcelの利用方法

3.1 PHPExcelのインストール

PHPExcelの最新版を参考文献⁵⁾に示したURLからDOWNLOADSを選択しダウンロードする。本稿執筆時には1.8.0 (2014/3/2リリース) が最新版である。マニュアルが必要ない場合はCode only

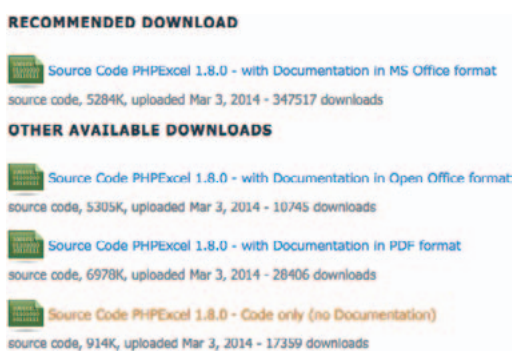


図1 PHPExcelのダウンロード

をダウンロードするとファイルサイズが小さく、時間がかからずに済む（図1）。

ダウンロードしたファイルはzipで圧縮されているため展開する。展開すると、Code onlyの場合はディレクトリClassesと3つのファイルが含まれている。PHPExcelの実行に必要なファイルはClasses内のファイルのみなので、プログラムを作成するディレクトリにこのディレクトリを移動する。ここでは、testPHPExcelと名付けたディレクトリで作業を行う（図2）。

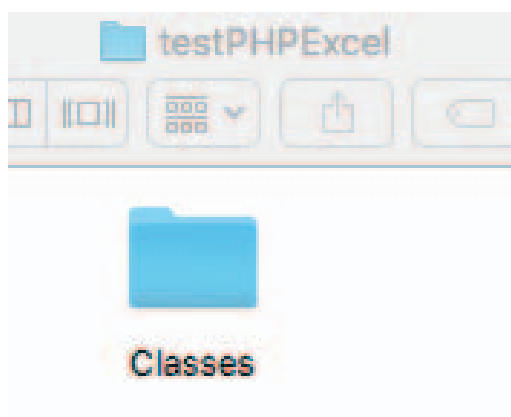


図2 PHPExcelのインストール

3.2 Excelファイルの読み込み

本節ではExcelを読み込むためのプログラムについて説明をする。全てのプログラムを一カ所に記載すると長いので分割して説明する。尚、プログラムは全てタグ“<?php,” “?”>”に囲まれているものとする。

3.2.1 読み込みの準備と読み込み

図3にExcelファイルを読み込むための基本的な設定と読み込み方法を示す。

1行目はライブラリを読み込むためのパスの設定を行う。Webアプリケーションとしてサーバサイドで利用する場合も、同様の設定を利用する。

2行目はTime Zoneの設定であり、PHPの設定状況によっては必要ない。

3～4行目が利用するライブラリの読み込みとなる。1行目で設定したパスに対しての相対的な参照となる。

5行目でファイルの形式を指定して、ファイルを読み込むためのインスタンス

```
1. set_include_path(get_include_path() . PATH_SEPARATOR . './Classes/');
2. date_default_timezone_set('Asia/Tokyo');
3. include_once("PHPExcel.php");
4. include_once("PHPExcel/IOFactory.php");
5. $objReader = PHPExcel_IOFactory::createReader("Excel2007");
6. $objReader->setIncludeCharts(TRUE);
7. $objPHPExcel = $objReader->load("readTest.xlsx");
```

図3 パスの設定とExcelファイルの読み込み

```

8.  $documentTitle = $objPHPExcel->getProperties()->getTitle();
9.  echo $documentTitle . "¥n";
10. $sheetsCount = $objPHPExcel->getSheetCount();
11. for($i = 0; $i < $sheetsCount; $i++){
12.     $objPHPExcel->setActiveSheetIndex($i);
13.     $sheetName = $objPHPExcel->getActiveSheet()->getTitle();
14.     echo "index:" . $i . "->" . $sheetName . "¥n";
15. }

```

図4 タイトルとシート名の取得

```

16. $objPHPExcel->setActiveSheetIndex(0);
17. $activeSheet = $objPHPExcel->getActiveSheet();
18. $cells = $activeSheet->getCellCollection();
19. foreach($cells as $cell){
20.     $objCell = $activeSheet->getCell($cell);
21.     $val = $objCell->getValue();
22.     $calculatedVal = $objCell->getCalculatedValue();
23.     $fcode = $activeSheet->getStyle($cell)->getNumberFormat()->getFormatCode();
24.     echo $cell . " -> " . $val . " : " . $calculatedVal . " : " . $fcode . "¥n";
25. }

```

図5 セル内のデータの取得

を取り出す。

6行目はグラフデータを読み出すことを指定している。

7行目でファイル” readTest.xlsx”を読み込む。

10行目ではシートの枚数を取得し、11

行目から15行目までの” for” 文によって全てのシート名の取得・表示を行っている。

3.2.3 セル内のデータの取得

3.2.2 プロパティとシートの確認

Excel ファイルのプロパティとシート数、シート名の確認方法は図4の通りである。

8行目でプロパティ内のタイトルを取出し、9行目で表示している。

図4までの操作でシートの情報を取得できる。次に指定したシートにおいてセル内のデータを取得する方法を図5に示す。

16行目で情報を取得したいシートを指定している。シートの番号は先頭から0、

```

26. $charts = $activeSheet->getChartCollection();
27. $chartNum = $activeSheet->getChartCount();
28. for($i = 0; $i < $chartNum; $i++){
29.     $chartDataSeries = $charts[$i]->getPlotArea()->getPlotGroupByIndex(0);
30.     echo $chartDataSeries->getPlotType() . "¥n";
31.     $plotValues = $chartDataSeries->getPlotValues();
32.     $count = count($plotValues);
33.     for($j = 0; $j < $count; $j++){
34.         echo $plotValues[$j]->getDataSource() . "¥n";
35.         $data = "";
36.         foreach($plotValues[$j]->getDataValues() as $datum){
37.             if($data == ""){
38.                 $data = $datum;
39.             } else {
40.                 $data .= "," . $datum;
41.             }
42.         }
43.         echo $data . "¥n";
44.     }
45. }

```

図6 グラフ情報の取得

1, 2, …となる。17行目でそのシートを取り出している。

18行目ではシート内のデータが入っているセルの一覧を取得している。

19行目から25行目で各セルに対してデータの取出しを行っている。

21行目でセルに入力されている値・式などを取り出している。22行目ではセルに入力された式を計算した結果が取得できる。例えば、セルA5に” = sum (A2:A4)” と入力されていた場合に、21行目では “= sum (A2:A4)” が取り出され、22行目ではセルA2からA4の値を合計した結果が得られる。

23行目ではセルに表示する際の書式を

取り出す。「標準」の場合は “General” が取得される。

24行目で各セルに対する結果を出力している。

3.2.4 グラフ情報の取得

最後にグラフに関する情報をどのように取得するかについて説明する（図6）。

26行目でグラフデータの集合を取得している。27行目はグラフの数を得る。

28行目から45行目にかけて各グラフについて情報を取り出している。

29行目でグラフを指定し、30行目でグラフの種類を出力している。グラフの種

類は以下のように出力される。

- 棒グラフ、横棒グラフ：barChart
- 折れ線グラフ：lineChart
- 円グラフ：pieChart
- 面グラフ：areaChart
- 散布図：scatterChart

31行目から44行目ではグラフ作成に使われているデータの場所と値を取得している。31行目でデータの参照範囲を全て取出し、32行目でそれがいくつあるかを計測している。

33行目から42行目にかけて各参照に対して調べている。34行目でどの範囲のデータを参照しているか出力する。36行目から43行目にかけてその参照による値を取り出している。具体的には以下のような情報を取り出すことになる。

34行目：'Sheet1'!\$A\$2:\$A\$4

43行目：2, 8, 4

3.3 実行例

図7で示したExcelファイルに対して図3～図6のプログラムを実行する。セルのA2, A3, A4には数値データが入っている。A5には“=SUM (A2:A4)”が入力されており、この3つのセルの合計値が表示される。また、折れ線グラフもこの3つの値に対して作成をした。プロパティ内のタイトルは「PHPExcelテスト」としている。実行例を図8に示す。

実行例において最初にプロパティ内の

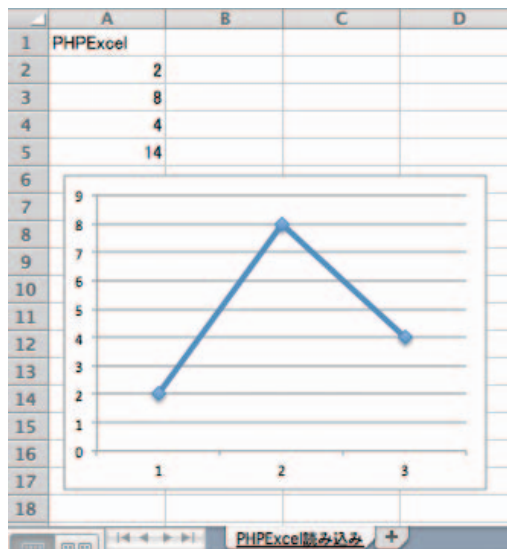


図7 対象のExcelファイル

```
[testPHPExcel] php readExcel.php
PHPExcelテスト
index:0->PHPExcel読み込み
A1 -> PHPExcel : PHPExcel : General
A2 -> 2 : 2 : General
A3 -> 8 : 8 : General
A4 -> 4 : 4 : General
A5 -> =SUM(A2:A4) : 14 : General
lineChart
PHPExcel読み込み!$A$2:$A$4
2,8,4
```

図8 実行例

タイトルが出力されている。次に現在参照しているシート名が表示される。その後、各セル内の情報が取り出され、最後の3行はグラフに関する情報となっている。セルA5の情報については、前述のように最初に入力された式が得られ、その次に計算結果が得られる。このようにPHPExcelを使うことでExcelファイルからいろいろな情報を抜き出すことが出

来る。

4. まとめ

本稿ではPHPExcelの基本的な使い方の説明をした。PHPのインストール、PHPExcelのインストールの仕方も示した。そして、Excelファイルを読み込むためのサンプルプログラムを記載し、その実行例を示した。

PHPExcelはExcelファイルを読み込むだけでなく、新規作成・編集をプログラムから行うことが出来る。次稿ではその方法について説明をしたい。また、今回はコマンドラインでの実行方法について示したが、Webアプリケーションとしてサーバサイドで実行する場合の設定等についても述べていきたい。

参考文献

- 1) 岩田員典, 功刀由紀子, 齋藤毅, 谷口正明, 長谷部勝也, 松井吉光, 古川邦之: Excel, Word自動採点システムHITsの構築と運用, 愛知大学 情報メディアセンター紀要「COM」, Vol.20, No.1 第35号, pp.11-23 (2010).
- 2) 岩田員典, 松井吉光, 長谷部勝也, 谷口正明, 池森均, 梅垣敦紀, 齋藤毅, 澤田貴行, 土橋喜, 中尾浩, 西本寛, 古川 邦之, 毛利元昭: 情報リテラシーのためのWord, Excel自動採点システムの構築と運用, 教育改革ICT戦略大会 pp.294-295 (2013).

- 3) SpreadSheet-ParseExcel,
<http://search.cpan.org/dist/Spreadsheet-ParseExcel/lib/Spreadsheet/ParseExcel.pm>
- 4) Standard ECMA-376 Office Open XML File Formats,
<http://www.ecma-international.org/publications/standards/Ecma-376.htm>
(2006-2012).
- 5) PHPExcel,
<https://phpexcel.codeplex.com>
- 6) PHP,
<http://php.net>
- 7) PHP For Windows,
<http://windows.php.net/download>

